Path Planning for Grasping Operations Using an Adaptive PCA-based Sampling Method

Jan Rosell, Raúl Suárez and Alexander Pérez

Abstract The planning of collision-free paths for a hand-arm robotic system is a difficult issue due to the large number of degrees of freedom involved and the cluttered environment usually encountered near grasping configurations. To cope with this problem, this paper presents a novel importance sampling method based on the use of Principal Component Analysis (PCA) to enlarge the probability of finding collision-free samples in these difficult regions of the Configuration Space with low clearance. By using collision-free samples near the goal, PCA is periodically applied in order to obtain a sampling volume near the goal that better covers the free space, improving the efficiency of sampling-based path planning methods. The approach has been tested with success on a hand-arm robotic system composed of a four-finger anthropomorphic mechanical hand (17 joints with 13 independent degrees of freedom) and an industrial robot (6 independent degrees of freedom).

Keywords Importance sampling, Principal Component Analysis, Anthropomorphic hands, Motion planning, Grasping

1 Introduction

Robotics is continually broadening its field of application, mainly towards service robotics, following advances in all of its disciplines. The improvement of manipulation capabilities is decisively contributing to this

The authors are with the Institute of Industrial and Control Engineering - Universitat Politècnica de Catalunya, Barcelona, Spain jan.rosell@upc.edu. A. Pérez is also with the Escuela Colombiana de Ingeniería "Julio Garavito", Bogotá D.C., Colombia.

tendency. To this end hand-arm robotic systems are being developed not solely within the scope of humanoid robotics but also for mobile manipulators. There are anthropomorphic mechanical hands with a number of degrees of freedom (*dof*) ranging from 12 (four fingers with 3 independent *dof* each one) to 25 (five fingers with 4 independent *dof* each one plus some *dof* in the palm) (Bicchi, 2000; Biagiotti et al, 2004). Therefore, hand-arm robotic systems are complex mechanisms with many degrees of freedom, and the automatic determination of their movements is difficult due to the high dimensionality of the corresponding Configuration Space ($\mathcal{C}$space). For this reason, several approaches were proposed to reduce the dimensionality of the problem, in particular determining correlations between the movements of the hand joints, to look for grasping configurations (Santello et al, 1998; Ciocarlie and Allen, 2009), for the path planning stage (topic discussed in detail below), as well as in relation with their influence in the grasping forces (Gabiccini et al, 2011). Moreover, robustness in front of uncertainty and sensory-motor coordination issues is a topic that has also been considered for the execution in real environments (Hsiao et al, 2011; Laschi et al, 2008).

To cope with high-dimensional path planning problems, sampling-based approaches have been proposed. These methods avoid the explicit characterization of the $\mathcal{C}$space, requiring only the collision evaluation of a discrete set of sample configurations and their interconnection with simple collision-free paths (Choset et al, 2005). Despite its simplicity, these methods have successfully solved many difficult problems involving a large number of degrees of freedom, being its efficiency tied to the capability of sampling those regions of the $\mathcal{C}$space relevant to the query to be solved, i.e. the sampling procedure is a key factor of this kind

of planners. Different importance sampling strategies have been proposed towards this end (Geraerts and Overmars, 2004; Hsu et al, 2006), like those that over-sample the $\mathcal{C}$space but quickly filter any non-promising configuration, e.g. (Boor et al, 1999; Hsu et al, 2003), or those that bias the sampling using the information gathered during the construction of the roadmap or tree, e.g. (Kavraki et al, 1996; Hsu et al, 2005). In order to improve the performance of sampling-based planners, dimension-reduction techniques have also been proposed, e.g. by using information provided by the user or by the constraints of the task (Berenson et al, 2009; Stilman, 2010), or by capturing the coupling that there may exist between the degrees of freedom of the mechanism using Principal Component Analysis (Safonova et al, 2004; Rosell et al, 2009).

Principal Component Analysis (PCA) has also been used to bias sampling, as first proposed by Dalibard and Laumond (2008) to accelerate the diffusion of a Rapidly-exploring Random Tree (RRT) within a narrow passage. The approach modifies the traditional extension step of the RRT algorithm by applying the PCA to a set of neighbors of the node to be extended, in this way the direction of maximum variance of the growing tree is obtained and it is then used to change the traditional isotropic sampling into a ellipsoidal sampling that emphasizes the directions of maximum growing of the tree. The approach locally estimates the direction of passages in order accelerate the diffusion within them. These authors also proposed the use of PCA for the expansion step of a Probabilistic RoadMap planner (PRM) (Dalibard and Laumond, 2011), i.e. they selected nodes in difficult areas as proposed by Kavraki et al (1996) and applied the PCA to its neighbor nodes to expand it as done with the extension step of the RRT. The use of PCA within a PRM was simultaneously proposed by Rosell et al (2011a), to determine a sampling region that attempts to tightly bound the free space of a difficult area of the $\mathcal{C}$space, like a narrow passage. This approach does not require a PCA computation per new sample, although it requires the specification of the region where the narrow passage might approximately lie.

Based on the approach of Rosell et al (2011a), the present paper proposes a PCA-based PRM to plan the motions of a hand-arm robotic system, in particular the motions close to the final grasping configurations, where the environment is usually cluttered and the solution paths have low clearances. Besides this use of PCA to bias samples, the planner proposed here also uses PCA as a dimension-reduction technique to obtain human-like motions (Suárez et al, 2009).

The paper is structured as follows. Sections 2 and 3 deal, respectively, with the PCA method as a dimension-reduction technique and as an importance sampling method, and Section 4 proposes a planner that incorporates both approaches for the planning of a hand-arm robotic system composed of an industrial robot and an anthropomorphic mechanical hand. Finally, Section 5 presents the conclusions of the work.

2 Principal Component Analysis

PCA is a statistical technique used to process a set of vectorial samples in order to look for a new orthogonal base of the vectorial space whose axis indicate, in a decreasing order, the directions of the space with more information to discriminate the samples, i.e. the dispersion of the samples is maximal along the first direction of the new base and decreases along the remaining ones. PCA is a common preprocessing step used to simplify the problem in pattern recognition and classification applications as well as in compression schemes and, in the field of motion and path planning, it is frequently used to reduce the dimension of the searching space and therefore to decrease the running time of the planning procedures.

There are different ways of performing the PCA (Jolliffe, 2002). Basically, it can be done by computing the eigenvalue decomposition of a data covariance matrix or the singular value decomposition of a data matrix, usually after mean centering the data for each attribute. The larger the eigenvalues or the singular values the larger the dispersion of the data along the corresponding eigenvector direction; the eigenvectors are directly used to define the directions of the new base.

2.1 Dimension reduction using PCA

PCA is used to reduce the dimension n of the initial space of samples, using instead a subspace of dimension $m < n$ defined by the first m directions of the new base obtained with the PCA and neglecting the others. Fig. 1 shows a simple illustrative example of the use of PCA to reduce the dimension of the space of samples $\mathcal{C}$. The grey dots represent samples in a 2-dimensional space defined by the original variables x_1 and x_2 (which may represent two real features of the problem). O' represents the mean of the set of samples, so the samples are first modified as $\mathbf{x}' = \mathbf{x} - O'$. Then, using PCA, a new base defined by x'_1 and x'_2 (which could be considered two virtual features) determines a new reference system with the origin at O'. Now, since the dispersion of the samples is larger along x'_1, the component x'_2 is
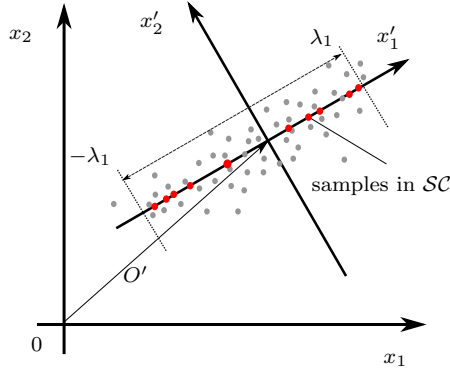
Fig. 1: Illustration of the use of PCA as a dimension reduction technique. The original 2-dimensional space $\mathcal{C}$ is defined by x_1 and x_2, the mean of the set of samples is O', the direction with maximal dispersion of samples is given by x_1', and the new base is defined by x_1' and x_2'. Then, the new working subspace is defined by x_1', and the valid portion is constrained to the range $[-\lambda_1, \lambda_1]$.

neglected, which is equivalent to consider the subspace $\mathcal{SC} \subset \mathcal{C}$ defined only by x_1' as the working space instead of $\mathcal{C}$, so the dimension was reduced from 2 to 1. Finally, a portion of $\mathcal{SC}$ defined by the range $[-\lambda_1, \lambda_1]$, such that it includes a desired percentage of the original samples, is used for the generation of new samples.

2.2 Principal Motion Directions based on PCA and its application in planning

The directions determined by the axis x_i' of the base obtained with the PCA were called Principal Motion Directions (PMDs) by Rosell et al (2009). Ordering the PMDs according to the dispersion along the axis x_i', the range of motion along the first PMD is the largest, and successively decreases for the rest. Therefore, allowing movements along the first $m \ll n$ PMDs will likely cover a significant portion of the valid $\mathcal{C}$space. Based on this, the PMDs were used in path planning for a hand-arm systems as follows. First, the natural workspace of a human operator's hand is sampled using a sensorized glove, and these samples are mapped to the configuration space of a mechanical hand making the resulting postures as similar as possible to those of the operator hand. Then, a PCA is performed in the mechanical hand configuration space in order to obtain the corresponding PMDs. Finally, for a given task, a solution path is planned using a PRM in a $\mathcal{C}$space with reduced dimensionality obtained considering only a reduced number of PMDs of the hand together with the six degrees of freedom of the arm (since the initial and goal configurations may be outside the reduced workspace they are specifically connected to the PRM).
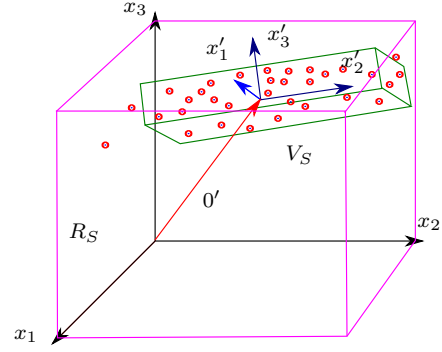

Fig. 2: Region R_S and sampling volume V_S obtained using PCA for a three dimensional $\mathcal{C}$space.

A variable number (increasing, starting from one) of PMDs in the planning phase has been also considered (Suárez et al, 2009; Rosell et al, 2011b).

The approach proposed in this work follows this line, but, besides the dimension reduction, it proposes a new original use of the PCA to improve the sampling phase for the application of the PRM.

3 Importance Sampling based on Adaptive Principal Component Analysis

3.1 The key idea

The key idea of using PCA in an importance sampling method for a PRM is twofold: a) define a new basis for the sampling space able to generate with a greater probability collision-free configurations in difficult areas of the $\mathcal{C}$space (the whole new base is considered, i.e. no reduction of dimensionality is pursued); and b) periodically recompute this basis as new collision-free configurations are obtained, making the process adaptive and obtaining a continuous improvement of the sampling performance. The sampling procedure proposed is conceived as a local method, i.e. it is applied to a region of the $\mathcal{C}$space where the area of interest is known to be located (e.g. a region containing a narrow passage).

Let R_S and V_S be two regions defined as follows (Fig. 2):

- R_S: Region of the $\mathcal{C}$space where importance sampling is required.
- V_S: Hyper-box aligned with the eigenvectors resulting from applying PCA to the collision-free samples within R_S, and with the size of each side proportional to the corresponding eigenvalue.

Given a set of collision-free samples from R_S, the Principal Component Analysis is used to obtain V_S. Then, the sampling procedure samples configurations

Algorithm 1 PCA-based Sampling

Require:
R_S: region of $\mathcal{C}$space
S: set of at least d collision-free samples from R_S
k_{R_S}: number of free configurations to sample from R_S
k_{V_S}: number of free configurations to sample from V_S
Ensure:
S enlarged with up to $k_{R_S} + k_{V_S}$ new collision-free samples

V_S=PCA(S)
S_R = SAMPLE-FROM(R_S, k_{R_S})
$S = S \cup S_R$
for $i = 1$ to k_{V_S} **do**
    s = SAMPLE-FROM(V_S, 1)
    if $s \in R_S$ **then**
        $S = S \cup s$
    end if
end for
return S

from both regions R_S and V_S and stores them to update V_S in the next call to the algorithm. The reason behind keeping the sampling in R_S is the obtention of collision-free configurations not included in V_S that allow the recomputation of V_S to better cover the area of interest.

3.2 Procedure

Algorithm 1 describes the PCA-based sampling procedure. This is a variation of the algorithm proposed by Rosell et al (2011a) that is simpler and guarantees the obtention of collision-free configurations from both V_S and R_S. It uses the following functions:

- SAMPLE-FROM(B, n): Returns n collision-free configurations sampled from region B.
- PCA(S): Performs the Principal Component Analysis over the samples of the set S and returns an hyper-box aligned with the resulting new base, centered at the mean value of S, and with the length of each side proportional to the standard deviation of the data along the corresponding axis.

Given an initial set S of at least d collision-free configurations (d being the dimension of the $\mathcal{C}$space) and given a region R_S, the sampling procedure computes V_S using PCA and then samples k_{R_S} collision-free configuration from R_S and k_{V_S} collision-free configuration from V_S. The algorithm returns the set S enlarged with the new collision-free configurations that belong to R_S (note that V_S may not be completely inside R_S and therefore some configurations sampled from V_S may not pertain to R_S). Fig. 3 illustrates the sampling regions R_S and V_S.

Region V_S fits the difficult area of the $\mathcal{C}$space better than R_S, like oriented bounding boxes (OBBs) fit object volumes better than axis-aligned bounding boxes
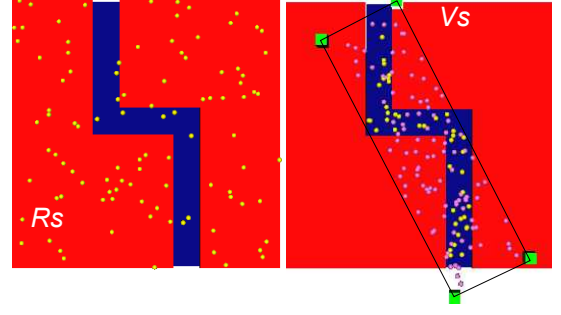

Fig. 3: Examples of samples obtained from regions R_S and V_S for a region with a narrow passage.

(AABBs) (Gottschalk et al, 1996). Therefore sampling in V_S may enlarge the probability of finding collision-free samples in low-clearance difficult regions of $\mathcal{C}$space. It must be noted, however, that the approach may become useless if the region V_S results (approximately) equal to the region R_S, but this situation only happens if the collision-free region has a (approximated) symmetric distribution inside R_S and its bounding box is similar to R_S. In grasping tasks, like in many others, this is a very improbable case, and, if for some particular application results $V_S \equiv R_S$ then the approach will run as a traditional regular PRM without any problem.

4 Application to the path planning of a hand-arm system

This section proposes a PRM planner that uses the PCA-based sampling method to solve path planning problems for a hand-arm robotic system. The proposed PRM is not conceived to capture the connectivity of the whole free space of the $\mathcal{C}$space, but solely the connectivity of the part that is relevant to connect two given configurations, c_{ini} and c_{goal}, being the later a grasp configuration. In the following subsections, the $\mathcal{C}$space of the problem and the regions where samples are to be obtained are defined; then the PRM is described and evaluated with simulated and real experiments.

4.1 The Configuration Space of the problem

Let $\mathcal{C}$ be the $\mathcal{C}$space of a hand-arm robotic system:

$$\mathcal{C} = \mathcal{C}^{\mathrm{a}} \times \mathcal{C}^{\mathrm{h}} \tag{1}$$

where $\mathcal{C}^{\mathrm{a}}$ and $\mathcal{C}^{\mathrm{h}}$ are, respectively, the $\mathcal{C}$spaces of the arm and of the mechanical hand. Using the Principal Motion Directions described in Section 2.2, the path planning will be done in a subspace $\mathcal{SC}$ defined as:

$$\mathcal{SC} = \mathcal{C}^{\mathrm{a}} \times \mathcal{SC}^{\mathrm{h}} \tag{2}$$

where $\mathcal{SC}^{\mathrm{h}}$ is the H-dimensional subspace of $\mathcal{C}^{\mathrm{h}}$ defined by the first H PMDs. Then, if A is the dimension of $\mathcal{C}^{\mathrm{a}}$, the planning will be done in a d-dimensional space with:

$$d = A + H.$$

Therefore, a configuration $c \in \mathcal{SC}$ will be a d-dimensional vector whose first A components correspond to the joints of the arm and whose last H components correspond to the H PMDs used to determine the values of the hand joints.

4.2 Sampling regions

This subsection defines the subregions of $\mathcal{SC}$ where to obtain samples for the proposed PRM. Let:

- $\mathrm{pos}(c)$ be the function that returns the position coordinates of the arm wrist when the hand-arm system is located at configuration c,
- $\mathrm{dist}(p_1, p_2)$ be the function that computes the Euclidean distance between two points $p_1, p_2 \in \mathbb{R}^3$,
- B be a region of $\mathcal{SC}$ defined as:

$$B(p, \delta) = \{c \in \mathcal{SC} \mid \mathrm{dist}(\mathrm{pos}(c), p) \leq \delta\}, \qquad (3)$$

  with $p \in \mathbb{R}^3$ and δ being a given distance threshold,
- $\mathrm{ConnComp}(c_{goal})$ be the set of configurations pertaining to the connected component of the PRM that contains c_{goal}.

Then, the following regions are defined:

- Region R_S. Usually, the paths of a hand-arm system to grasp an object have low clearances near the goal grasp configuration. Therefore, it is near the goal configuration where the PCA-based sampling proposal better contributes to improve sampling-based planners. For this reason, R_S is defined as the region of $\mathcal{SC}$ containing the positions of the arm (i.e. the x, y and z coordinates of the robot wrist) whose distance from the position of the arm when it is located at the goal grasp configuration c_{goal} is below a given threshold δ_R, i.e.:

$$R_S(\delta_R) = B(\mathrm{pos}(c_{goal}), \delta_R) \qquad (4)$$

  A practical value of δ_R is chosen approximately equal to the finger lengths.
- Region V_S. It is the region resulting from the PCA applied to the set S of all collision-free hand-arm configurations within R_S that belong to the same connected component than c_{goal}. V_S is a hyper-box of dimension d, aligned with the resulting new base, centered at the mean value of S, and with the length

of each side equal to three times the standard deviation of the data along the corresponding axis. Sampling within V_S results in hand-arm configurations $(\theta_1, \ldots, \theta_A, \ldots, \theta_{A+H}) \in \mathcal{SC}$.

- Region I_S. In order to compute V_S for the first time, a set of at least d samples are required (i.e. a number equal to the dimension of the $\mathcal{C}$space). These are obtained by sampling a region I_S defined around c_{goal} as:

$$I_S(\delta_I) = B(\mathrm{pos}(c_{goal}), \delta_I), \qquad (5)$$

  with δ_I being a threshold smaller than δ_R (a practical value δ_I is around a third of δ_R). Sampling from I_S is done as follows:
  - Arm configuration: The arm position is obtained by sampling a sphere of radius δ_I centered at $\mathrm{pos}(c_{goal})$; the arm orientation is obtained by sampling around the goal orientation (the orientation is parameterized with three parameters using quaternions (Kuffner, 2004), and these parameters are varied a small amount around the values corresponding to c_{goal}). Afterwards, the arm configuration $(\theta_1, \ldots, \theta_A) \in \mathcal{C}^{\mathrm{a}}$ is obtained using the inverse kinematics (if there are multiple solutions the same arm configuration as in c_{goal} is chosen).
  - Hand configuration: The hand configuration $(\theta_{A+1}, \ldots, \theta_{A+H}) \in \mathcal{SC}^{\mathrm{h}}$ is obtained by sampling each PMD in a predefined range around c_{goal}.
- Region M_S. Samples further away from c_{goal} than those of I_S and V_S are needed to construct the whole roadmap to solve the query to connect c_{ini} and c_{goal}. Also, collision-free samples outside V_S are needed to recompute V_S to improve the coverage of the area of interest. For these purposes, a region M_S is defined in a similar way as R_S and I_S, but centered at any configuration of the connected component of c_{goal} and with a variable distance threshold that ranges from δ_I to the distance between the initial and the goal arm positions, i.e.:

$$M_S(\delta_M) = B(\mathrm{pos}(c_{cg}), \delta_M) \qquad (6)$$

  with $\delta_I \leq \delta_M \leq \mathrm{dist}(\mathrm{pos}(c_{ini}), \mathrm{pos}(c_{goal}))$ and $c_{cg} \in \mathrm{ConnComp}(c_{goal})$. Sampling from M_S is done as follows:
  - Arm configuration: The arm configuration is obtained as done from I_S, but slightly changing the procedure for the arm position, which is obtained by, first, randomly selecting a configuration c_{cg} of the connected component of c_{goal}, and then sampling a sphere of radius δ_M centered at $\mathrm{pos}(c_{cg})$.

– Hand configuration: The hand configuration is obtained by sampling each PMD within its whole range.

4.3 PCA-based PRM

Algorithm 2 describes the proposed PRM that uses the PCA-based sampling method to solve problems for a real hand-arm robotic system; it is called PCA-based PRM. The algorithm first obtains collision-free configurations around the goal configuration by sampling within I_S, then it iteratively executes the following three steps until the solution is found or a maximum number of samples have been generated:

1. (Re)computes V_S using the samples of the roadmap within R_S.
2. Samples from V_S, adds the samples to the roadmap, searches for a solution and exits if it is found.
3. Samples from M_S, adds the samples to the roadmap, searches for a solution and exits if it is found.

Steps 2 and 3 are done by Algorithm 3, which use the following functions:

– SAMPLE-FROM(B, n): Returns n collision-free configurations sampled from region B, following the steps described in the previous subsection for regions I_S, V_S and M_S.
– PCA(S): Is the same function introduced in Section 3.2, with the length of each side of the returned hyperbox equal to three times the standard deviation of the data along the corresponding axis.
– PRM-ADD(S): Builds a roadmap with the set of samples S.
– PRM-SOLVE(s): Adds the collision-free configuration s to the roadmap and returns the path connecting the initial to the goal configuration, if it exists, or the empty-set otherwise.
– CONN-COMP(s, c): Returns true if s and c pertain to the same connected component of the roadmap.

In order to explore the part of the $\mathcal{C}$space relevant to the query to be solved and to make emphasis around the cluttered zone around c_{goal}, the parameter δ_M defining the size of M_S is not fixed, but it is initially set as $\delta_M = \delta_I$ and then it is incremented at each iteration in steps δ_I (i.e. $\delta_M = \delta_M + \delta_I$), restarting at $\delta_M = \delta_I$ whenever δ_M becomes greater than $\mathrm{dist}(\mathrm{pos}(c_{ini}), \mathrm{pos}(c_{goal}))$.

The distance thresholds δ_R and δ_I depend on the mechanical hand used. Therefore, the planner has only two extra parameters to be defined by the user (the number of samples k_{M_S} and k_{V_S}), besides the parameters of a basic PRM like the maximum number of neighbors per node and the neighbor distance threshold. The best values of k_{M_S} and k_{V_S} are discussed in Section 4.5.

Algorithm 2 PCA-based PRM

Require:
  c_{ini}: Initial configuration
  c_{goal}: Goal configuration
  d: Dimension of the $\mathcal{C}$space
  δ_I, δ_R: Distance thresholds
  k_{M_S}: number of free configurations to sample from M_S
  k_{V_S}: number of free configurations to sample from V_S
Ensure:
  P: path connecting c_{ini} and c_{goal}

  $S_{PCA} = $ SAMPLE-FROM(I_S, d)
  PRM-ADD(S_{PCA})
  $\delta_M = \delta_I$
  $trials = 0$
  while $trials < MaxTrials$ **do**
    $trials = trials + k_{M_S} + k_{V_S}$
    V_S=PCA(S_{PCA})
    $[P, S_{PCA}]$ =Sample-and-Solve(c_{ini},c_{goal},V_S,k_{V_S},S_{PCA})
    if $P \neq \emptyset$ **then**
      return P
    end if
    $[P, S_{PCA}]$ =Sample-and-Solve(c_{ini},c_{goal},$M_S(\delta_M)$,k_{M_S},S_{PCA})
    if $P \neq \emptyset$ **then**
      return P
    end if
    $\delta_M = \delta_M + \delta_I$
    if $\delta_M > \mathrm{dist}(\mathrm{pos}(c_{ini}), \mathrm{pos}(c_{goal}))$ **then**
      $\delta_M = \delta_I$
    end if
  end while
  return $\emptyset$

Algorithm 3 Sample-and-Solve

Require:
  c_{ini}: Initial configuration
  c_{goal}: Goal configuration
  Q: Sampling region
  k_Q: Number of samples
  S_{PCA}: Set of samples
Ensure:
  P: Path connecting c_{ini} and c_{goal}
  S_{PCA}: enlarged set of samples

  for $i = 1$ to k_Q **do**
    $s = $ SAMPLE-FROM(Q, 1)
    $P = $ PRM-SOLVE(s)
    if $P \neq \emptyset$ **then**
      return $[P, S_{PCA}]$
    end if
    if $s \in R_S$ **and** CONN-COMP(s, c_{goal}) **then**
      $S_{PCA} = S_{PCA} \cup s$
    end if
  end for
  return $[\emptyset, S_{PCA}]$

4.4 Implementation

The proposed approach has been implemented as a specialized PRM planner inside the home-developed path planning framework called *The Kautham Project* (Pérez and Rosell, 2010). This tool was developed with the

open-source and cross-platform directives in mind (Pérez and Rosell, 2009), and it uses libraries such as Qt (Blanchette and Summerfield, 2006) for the user interface, Coin3D (Kongsberg Oil & Gas Technologies, 2010) for the graphical rendering and PQP (Larsen et al, 2000) for the collision detection. This application provides the developer with direct and inverse kinematic models of the robots, and with samplers, metrics and other tools needed for the development of planners.

With respect to the PCA-based PRM introduced in the paper, there are several alternatives for its implementation. One possibility is to use Octave (Eaton, 2002) or R (Foundation, 2011) together with a package like the RCPP (Eddelbuettel, 2011) to connect them to the application. However, since our application requires to perform PCA within the sampling loop, the performance criteria is considered a key factor, and for this reason the Armadillo C++ Linear Algebra Library has been used (Sanderson, 2010). This library is also open-source and has a good performance in response time for large volumes of data (Rosell et al, 2011a).

4.5 Application Examples

Some application examples are given here to illustrate the ability of the proposed approach to find a solution path for a hand-arm robotic system moved towards a low-clearance grasp configuration. The hand-arm robotic system is composed of a TX90 Stäubli robot (6 *dof*) and the Schunk Anthropomorphic Hand, SAH (from which the first five PMDs have been selected). Therefore the dimension of the planning subspace $\mathcal{SC}$ is $d = 11$. In the first example the hand must grasp a thin T-shaped object and in the second one the hand must grasp a given (yellow) can in a cluttered environment. Fig. 4 and Fig. 5 show, respectively, an instance of the solution for each these two tasks.

Fig. 6 show the success rate for these two examples as a function of the values k_{M_s} and k_{V_s}, running 29 instances for each combination of values and using a maximum number of samples of 2,000 in the first example and 10,000 in the second one. It can be observed that the best rates occur for a large ratio k_{V_s}/k_{M_s}, showing the goodness of the proposal. In all the executions the distance thresholds for the SAH hand have been set to $\delta_R = 150$ mm and $\delta_I = 50$ mm.

The proposal has been compared with other importance sampling strategies:

- The Gaussian sampling strategy, that obtains configurations near the obstacles (Boor et al, 1999).
- The Bridge-test sampling strategy, that obtains configuration in narrow passages (Hsu et al, 2003).

Example	Approach	Samples	Success Rate	Time (s)
T-object	PCA	1,240	100%	43.1
	Gaussian	15,740	76%	1558.3
	Bridge-test	15,586	79%	1627.8
Yellow can	PCA	12,307	100%	269.9
	Gaussian	20,778	85%	1264.0
	Bridge-test	21,595	76%	1245.5

Table 1: Performance evaluation comparison. Average values for 72 executions of the tasks shown in Fig. 4 and 5 using different approaches (*Samples* refers to the mean number of configurations that have been collision-checked; *Time* is the mean time of the successful trials).

These methods have been chosen because they are a reference in many evaluations, e.g. (Hsu et al, 2005; Denny and Amato, 2013), and comparative studies, e.g. (Geraerts and Overmars, 2004; Hsu et al, 2006). In order to make a fair comparison between these methods, the Gaussian and the Bridge-test sampling methods have been implemented within the main procedure given in Algorithm 2, just substituting the sampling in V_S by the sampling in a region G_S defined as:

$$G_S(\delta_I) = B(\text{pos}(c_{cg}), \delta_I) \;\; \text{with} \;\; c_{cg} \in \text{ConnComp}(c_{goal}). \tag{7}$$

The sampling in G_S is implemented as follows:
a) For the Gaussian sampling:

- Instantiate G_S by randomly choosing a configuration $c_{cg} \in \text{ConnComp}(c_{goal})$.
- Obtain a sample s_1 from G_S as done from M_S.
- Generate a random sample s_2 at a distance from s_1 chosen according to a normal distribution.
- Between s_1 and s_2 return the one that is not in collision (if both are free or in collision then discard them).

b) For the Bridge-test sampling:

- Instantiate G_S by randomly choosing a configuration $c_{cg} \in \text{ConnComp}(c_{goal})$.
- Obtain two samples from G_S as done from M_S.
- If both are in collision then return the mid point if it is a free configuration, otherwise discard all of them.

The average performance results obtained are shown in Table 1, using a maximum number of samples of 40,000 in both examples. The proposed method run with $k_{V_s} = 10$ and $k_{M_s} = 1$ (i.e. a ratio 10:1), resulting a success rate of 100%, outperforming those of the Gaussian and of the Bridge-test methods. Besides, the number of samples required and the computational cost were also significantly smaller.
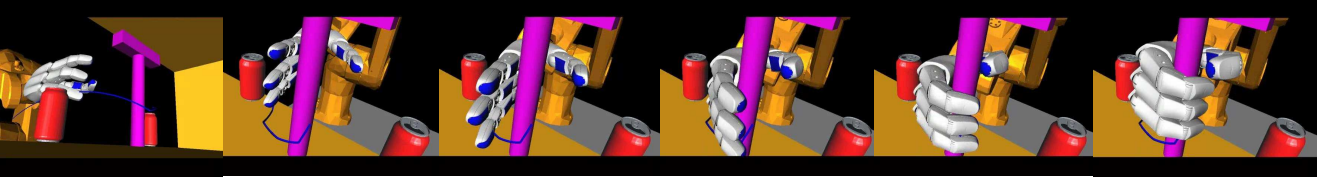
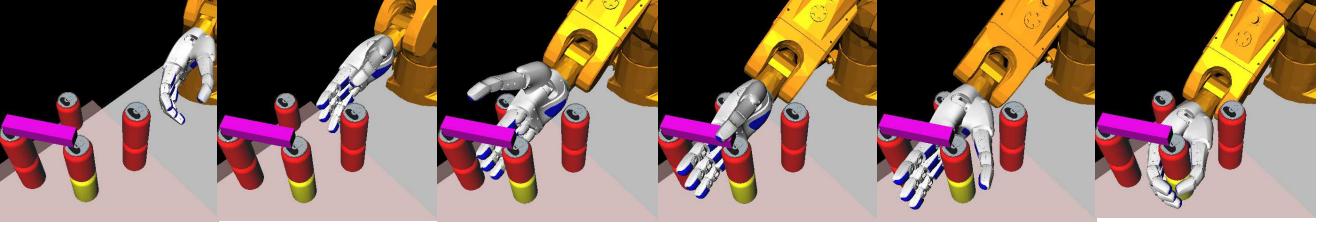Fig. 4: Simulation of the plan to grasp the T-shape object.

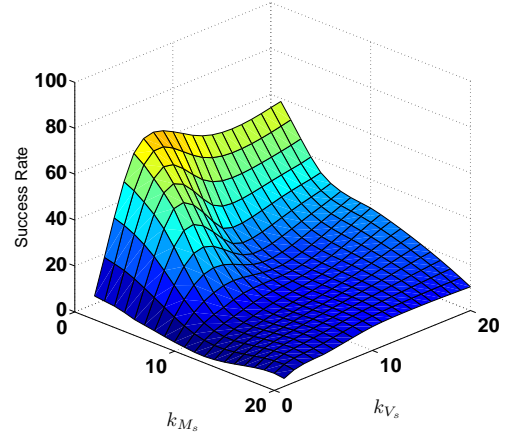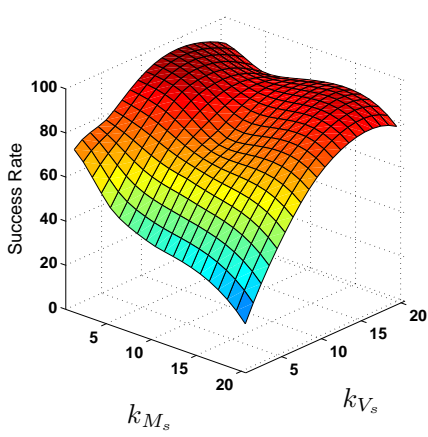Fig. 5: Simulation of the plan to reach the yellow can in a cluttered environment.

Fig. 6: Success rate as a function of k_{M_S} and k_{V_S} for the T-object example (left) and for the yellow can example (right). 29 instances of the problem were run for each combination of k_{M_S} and k_{V_S} in the range $[1, 20]$.

Fig. 7 illustrates, both in simulation and in the real environment, the solution path for a third example in which the goal is also to grasp the yellow can in a cluttered environment. The accompanying video shows the simulation and real execution of this task.

5 Conclusions

Principal Component Analysis has been previously used in the scope of path planning to reduce the dimensionality of the planning space. Within the scope of hand-arm robotic systems, PCA is also used in this paper as an importance sampling method, i.e. as a way to enlarge the probability to obtain samples from those difficult regions of the Configuration Space relevant to solve a query. The search of a collision-free path to reach an object to be grasped is a difficult issue due to the low clearances that there exist near the goal grasp configuration and the large number of degrees of freedom involved. A probabilistic roadmap path planner that uses the PCA-based sampling method near the goal grasp configuration has been proposed and tested with success on a hand-arm robotic system composed of a four-finger anthropomorphic mechanical hand and an industrial robot. As a future work we are planning to extend the approach to consider uncertainty on the pose of the object to be grasped, which may affect the free path of the hand during the final movements for the grasping operation.
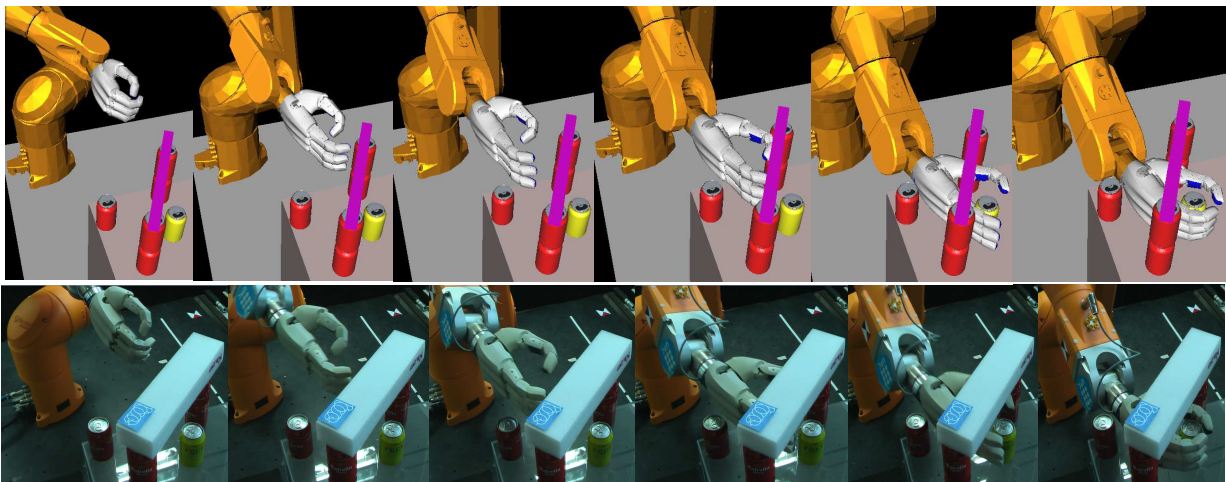
Fig. 7: Simulation and real execution of the plan to grasp the yellow can in a cluttered environment.

References

Berenson D, Srinivasa S, Ferguson D, Kuffner J (2009) Manipulation planning on constraint manifolds. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp 625 – 632

Biagiotti L, Lotti F, Melchiorri C, Vassura G (2004) How far is the human hand? a review on anthropomorphic robotic end-effectors. Tech. rep., University of Bologna

Bicchi A (2000) Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. IEEE Transactions on Robotics and Automation 16(6):652–662

Blanchette J, Summerfield M (2006) C++ GUI Programming with Qt 4. Prentice Hall PTR, Upper Saddle River, NJ, USA

Boor V, Overmars MH, van der Stappen AF (1999) The Gaussian sampling strategy for probabilistic roadmap planners. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp 1018–1023

Choset H, Lynch KM, Hutchinson S, Kantor G, Burgard W, Kavraki LE, Thrun S (2005) Principles of Robot Motion. The MIT Press

Ciocarlie MT, Allen PK (2009) Hand posture subspaces for dexterous robotic grasping. The International Journal of Robotics Research 28(7):851–867

Dalibard S, Laumond J (2008) Control of probabilistic diffusion in motion planning. In: Proc. of the Workshop on Algorithmic Foundations of Robotics, pp 467 – 481

Dalibard S, Laumond J (2011) Linear dimensionality reduction in random motion planning. The International Journal of Robotics Research 30(12):1461–1476

Denny J, Amato N (2012) Toggle PRM: A Coordinated Mapping of C-free and C-obstacle in Arbitrary Dimension. In: Frazzoli E, Lozano-Perez T, Roy N, Rus D (eds) Algorithmic Foundations of Robotics X, Springer Tracts in Advanced Robotics, vol 86, Springer Berlin Heidelberg, pp 297312

Eaton JW (2002) GNU Octave Manual. Network Theory Limited

Eddelbuettel D (2011) Rcpp: Seamless R and C++ integration. dirk.eddelbuettel.com/code/rcpp.html

Foundation TR (2011) The R project for statistical computing. www.r-project.org

Gabiccini M, Bicchi A, Prattichizzo D, Malvezzi M (2011) On the role of hand synergies in the optimal choice of grasping forces. Autonomous Robots 31:235 – 252

Geraerts R, Overmars MH (2004) Sampling techniques for probabilistic roadmap planners. Intelligent Autonomous Systems 8:600–609

Gottschalk S, Lin MC, Manocha D (1996) OBBtree: a hierarchical structure for rapid interference detection. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, New York, NY, USA, SIGGRAPH 96, pp 171180

Hsiao K, Pack-Kaelbling L, Lozano-Perez T (2011) Robust grasping under pose uncertainty. Autonomous Robots 31:253–268

Hsu D, Jiang T, Reif J, Sun Z (2003) The bridge test for sampling narrow passages with probabilistic roadmap planners. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp 4420–4426

Hsu D, Sanchez-Ante G, Sun Z (2005) Hybrid PRM sampling with a cost-sensitive adaptive strategy. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp 3874 – 3880

Hsu D, Latombe JC, Kurniawati H (2006) On the probabilistic foundations of probabilistic roadmap plan-

ning. The Intenational Journal of Robotics Research 25(7):627 – 643

Jolliffe I (2002) Principal Component Analysis. Springer Series in Statistics, Upper Saddle River, NJ, USA

Kavraki LE, Svestka P, Latombe JC, Overmars MK (1996) Probabilistic roadmaps for path planning in high - dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12(4):566–580

Kongsberg Oil & Gas Technologies (2010) Coin3D - 3d graphics development tools. www.coin3d.org

Kuffner J (2004) Effective sampling and distance metrics for 3d rigid body path planning. In: Proc. of Int. Conf. on Robotics and Automation, pp 3993 – 3998

Larsen E, Gottschalk S, Lin MC, Manocha D (2000) Fast proximity queries with swept sphere volumes. In: Proc. of Int. Conf. on Robotics and Automation, pp 3719–3726

Laschi C, Asuni G, Guglielmelli E, Teti G, Johansson R, Konodu H, Wasik Z, Carrozza M, Dario P (2008) A bio-inspired predictive sensory-motor coordination scheme for robot reaching and preshaping. Autonomous Robots 25:85–101

Pérez A, Rosell J (2010) The Kautham Project. URL https://sir.upc.es/kautham/.

Pérez A, Rosell J (2009) A Roadmap to Robot Motion Planning Software Development. Computer Applications in Engineering Education

Rosell J, Suárez R, Rosales C, García JA, Pérez A (2009) Motion planning for high DOF anthropomorphic hands. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp 4025–4030

Rosell J, Cruz L, Suárez R, Pérez A (2011a) Importance sampling based on adaptive principal component analysis. In: Proc. of the IEEE Int. Symposium on Assembly and Manufacturing

Rosell J, Suárez R, Rosales C, Pérez A (2011b) Autonomous motion planning of a hand-arm robotic system based on captured human-like hand postures. Autonomous Robots 31:87–102

Safonova A, Hodgins JK, Pollard NS (2004) Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. ACM Transactions Graph 23(3):514–521

Sanderson C (2010) Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Tech. rep., NICTA

Santello M, Flanders M, Soechting JF (1998) Postural hand synergies for tool use. Journal of Neuroscience 18(23):10,105–10,115

Stilman M (2010) Global manipulation planning in robot joint space with task constraints. IEEE Transactions on Robotics 26(3):576–584

Suárez R, Rosell J, Pérez A, Rosales C (2009) Efficient search of obstacle-free paths for anthropomorphic hands. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp 1773–1778